Instead, Use One Table of Function Pointers per Type

What if, **instead**,
- **for each type**
- **we create a table** (a `struct`+)
- filled **with function pointers**?

We only need **one such table per type**, NOT one table per variable.

And one pointer per variable—to the variable's type's table of function pointers.

+An array of pointers, but the functions have different signatures.

25

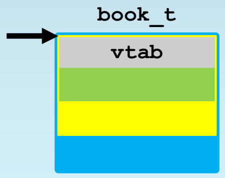
Virtual Function Table Pointer Refers to Table for Type

Let's **call the table pointer `vtab`**.+

And place it **at the start of the structure**, so that we can always find it.

Every book's `vtab` points to the table for books.

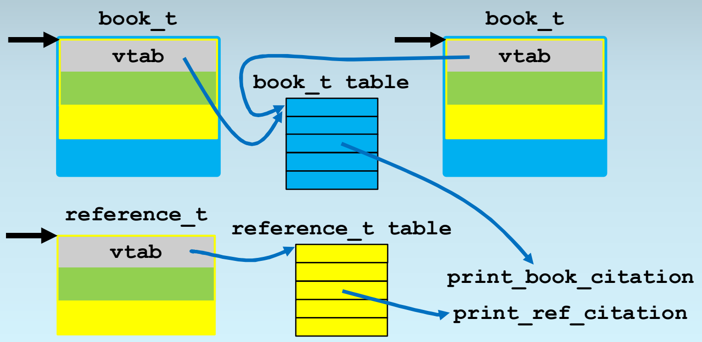
Every reference's `vtab` points to the table for references.



+For "virtual function table."

26

Calling a Virtual Function Requires Two Memory Reads



27

A Virtual Function Call Costs Extra Memory Reads

Two loads followed by a call (JSRR, for example), instead of just a call.

That's the more significant cost.

You may want to try converting our bibliography printing code into **LC-3** assuming that **`vtab`** is at the start of each structure (before the **`double_list_t`**). Choose some non-zero index for **`print_citation`** in the table to make it interesting (the index must be a constant, of course).

28